**Amendments to the Specification:**

Please make the following amendments to the application. For the ease of making these amendments, the specification as published will be used and the paragraph numbers set forth therein will be referred to herein:

Please replace paragraph [0002] with the following amended paragraph:

[0002] One computer implemented approach for computing demand forecast information for a demand forecast application involves defining a so-called "demand forecast tree" capable of being graphically represented by a single top level node (00) with at least two branches directly emanating therefrom, each branch having at least one node, for example, bottom level node (11) (see FIG. 1). The demand forecast information is computed on the basis of historical time series of observations. These observations are typically associated with bottom level nodes by a forecast engine commercially available, for example, from Demantra Ltd, Israel under the name Demantra ™ [[.TM.]] Demand Planner. One exemplary demand forecast application is the forecast of demand for a consumer item at an outlet ~~as represented by a bottom level node~~ on the basis of historical sales of the consumer item at the outlet. The outlet is represented by a bottom level node.

Please replace paragraph [0003] with the following amended paragraph:

[0003] Demand ~~Since demand~~ forecast information is becomingly increasingly important for enterprises for a myriad of planning and logistic purposes. One [[ , one]] important metric of a demand forecast application is its so-called "run time," which is [[time"]] defined as the interval between the start time of computing demand forecast information and its end time. For many enterprises, the run time of their demand forecast applications is becoming intolerably long due to increasingly complicated demand forecast trees, the number and sophistication of available mathematical models for forecasting purposes, and the like. Therefore, there is a need to shorten run time, but ~~time but~~ without militating against other equally important metrics of a demand forecast application, for example, accuracy, robustness, and the like.

Please replace paragraph [0007] with the following amended paragraph:

[0007] (c) simultaneously computing demand forecast information for at least two branches of the demand forecast tree on two different ones of said computer servers ~~of the at least two computer servers~~.

Please replace paragraph [0008] with the following amended paragraph:

[0008] The present invention is based on the realization that the demand forecast information for the top level node ~~(0,0)~~ (00) of a demand forecast tree is too abstract to be truly meaningful. Therefore, ~~meaningful, and therefore~~ a demand forecast tree can be effectively regarded as consisting of a number of sub-trees equal to the number of its branches directly emanating from its top level node ~~(0,0)~~ (00). By virtue of this, the demand forecast information for different branches or sub-trees of the same demand forecast tree can be computed independently from one another. This then enables ~~, thereby enabling~~ a forecast engine having two or more computer servers to shorten the run time to compute the demand forecast information for an entire demand forecast tree in comparison to that hitherto achievable by a single computer server. The computer servers of a forecast engine may be co-located or interconnected, for example, over an enterprise's Local Area Network (LAN), over its Wide Area Network (WAN), and the like. Thus, in accordance with the present invention, the run time is effectively delimited between the start time of the first computer server of a forecast engine to start computing demand forecast information for a demand forecasting application and the end time of its last computer server to finish computing demand forecast information for the same demand forecasting application. The present invention is particularly beneficial for demand forecast applications represented by wide demand forecast trees as opposed to narrow demand forecast trees. Wide demand forecast trees have ~~having~~, say, five or more branches since a greater number of computer servers can

simultaneously compute the demand forecast information ~~as opposed to narrow demand forecast trees~~. The present invention can be applied to new computerized demand forecast applications whilst existing computerized demand forecast applications can be readily retrofitted to support the present invention.

Please replace paragraph [0009] with the following amended paragraph:

[0009] Ideally, the two or more computer servers of a forecast engine simultaneously computing demand forecast information for a run of a demand forecast application would all finish computing simultaneously, thereby optimally minimizing run time. However, computer servers available for a run of a demand forecast application inter alia typically have different computing strengths, and are often not wholly dedicated to a run. Moreover, different branches of the same demand forecast tree typically have different numbers of bottom level nodes which largely determines the computing time required to compute demand forecast information. In view of the above, and since the actual computing time for computing demand forecast information for a branch of a demand forecast tree typically involves considerable computing overhead. The present invention provides for the case in which ~~, in the case that~~ the number of branches of a demand forecast tree exceeds the number of computer servers available for computing demand forecast information. Rather ~~, rather~~ than being processed separately, the branches of a demand forecast tree are preferably grouped into discrete tasks, each task ~~each~~ containing one or more

branches which are then available for allocation to computer servers for processing on a first come first served basis. The number of tasks is generally an integer multiple of the number of computer servers available for computing demand forecast information as specified by a user, and the computing demands of different tasks of a demand forecast application are preferably equalized as much as possible to facilitate minimizing its run time.

Please replace paragraph [0018] with the following amended paragraph:

[0018] FIG. 2 shows a table 2 containing historical time series of sales for one ~~each~~ of the bottom level nodes for the sales on various dates of item number 1, namely at node (11). ~~(11),~~ The other bottom nodes, namely nodes (12), . . . (43), and (51), also have corresponding historical time series of sales (not shown). ~~the~~ The unique key of each sales record being the Item ID (which is identified in the table as only the word "ITEM"), the Location ID (which is identified in the table as only the word "DATE"), and the Date. The number of sales then for a particular key is given under the column headed by the word "VALUE". For example, the number of sales of item number 1, which is represented by node (10), at a first location, represented by bottom node (11), on November 17, 2001 was 14, and on November 20, 2001 was 11. Similarly, the sales of item number 1 at a second location, represented by bottom node (12), are recorded in table 2 for the appropriate date, but are not shown in FIG. 2. Also, similarly, the sales for item number 2, which is represented by node 20, are recorded in table 2 for the corresponding

locations, which are represented by nodes (21) and (23), but are not shown in FIG. 2..

Please replace paragraph [0018] with the following amended paragraph:

[0019] FIG. 3 shows a computer implemented system 3 including a database server 4 for storing time series of sales, a forecast engine 6, for example, commercially available, for example, from Demantra Ltd. Israel under the trademark ~~name~~ Demantra ™ [[.TM.]] and the item Demand Planner. Forecast engine 6 includes ~~, including~~ two or more computer servers 7 each independently capable of computing demand forecast information for an entire branch of the demand forecast tree 1, and a computer manager 8 for allocating branches into tasks.

Please replace paragraph [0022] with the following amended paragraph:

[0022] In the event that CS=1, the computer manager 8 would assign all the branches A, B, C, D and E to the singly available computer server 7. Thus, computer server 7 would process all the branches in a similar manner to a conventional computer implemented system without the benefit of the present invention. In the event that the number of computer servers is greater than the number of branches, CS>BRANCH, the computer manager 8 would assign the branches A, B, C, D and E to five different tasks such that each branch or task is processed by a dedicated computer server 7. In the event that the number of branches is greater than the number of computer

servers, BRANCH>CS>1, such as three computer servers for five branches as shown in FIG. 6, the computer manager 8 typically divides the number of branches into the number of tasks given by the product of the values of CS and MULTIPLIER such that the expected computing time for each task is substantially equal. For example, assuming that CS=3 and MULTIPLIER=1, the computer manager 8 would group the branches A, B, C, D and E into three tasks as follows: Task 1 having branches A and B with a total of four bottom level nodes (11), (12), (21), and (23); Task 2 having branch C only with three bottom level nodes (31), (32) and (33); and Task 3 having branches D and E also with four bottom level nodes (41), (42), (43), and (51). Thus, the demand forecast tree 1 is effectively converted into three separate sub-trees 11, 12, and 13 (see FIG. 6).